Center for Teaching, Research and Learning
Research Support Group
American University, Washington, D.C.
Hurst Hall 203
rsg@american.edu
(202) 885-3862

Introduction to SAS

Workshop Objective

This workshop is designed to give a basic understanding of how SAS procedure works and how to use simple statistical methods to analyze data.

Learning Outcomes

1. Understanding the basics
2. Managing data
3. Computing descriptive statistics
4. Creating basic graphs
5. Running linear regressions

Scenario:

We are interested in studying the relationship between the Gross Domestic Product (GDP) per capita of a country and how corrupt the country is. To start investigating this, we will examine the relationship between nations' scores on the "Corruption Index" and their "GDP per capita".

I. Understanding the basics

In this section, we introduce a few basic but very helpful commands.

Brief Intro to the SAS Interface:

Editor Window

The Editor window is the place where the user programs the SAS procedures. It is equivalent to the do file in Stata or the syntax file in SPSS. The user can run a specific parts of the code by highlighting them and pressing F5 or clicking on the "RUN" button.

Explorer Window

The explorer window allows the user to browse through the active libraries and the content of the system.

Output Window

The output window contains the output from the various procedures

SAS Procedures

A SAS procedure is a collection of lines of code that carry out the statistical analysis. The procedure is always preceded by the keyword proc and followed by the name of the procedure. Once all statements have been included in the body of the procedure, the procedure is "finalized" with an end statement.

SAS Data Steps

This is the part of the code where the data is created/imported.

Importing Data

There are several ways in which the user can import a data set. In this example we illustrate two ways. To import a data set, the user can use the built-in import wizard by clicking on the following:

File > Import Data… > (select Comma Separated Values) > Next > Browse (to the location of the file) > Next > Type in the name of your SAS data set

Alternatively the user can use the **proc import** procedure in SAS. This can be done by typing the following code in the Editor window:

```
libname SASINTRO "C:\data\";

PROC IMPORT OUT= SASINTRO.nations
DATAFILE="J:\CLASSES\Workshops\cs.dta"
      DBMS=dta REPLACE;
RUN;
```

The SASINTRO library name is map to "C:\" drive. This can be map to any desired folder. The **out** parameter specifies the library where the data set should be saved. The **DBMS** parameter specifies the type of the dataset. In this case, we are importing a STATA dataset (.dta).

Program-File

We recommend that you create a program file before you start working in SAS. A Program-file allows you to record and save all of your command lines so that you can repeat those steps in the future.

- **File, New Program**
- **Save Program File to G:\ or any other drive**

II. Descriptive statistics

In this section, we will demonstrate how to obtain information on descriptive statistics for one or more variables.

A full description of all variables in the data can be obtained with the *print/contents* procedure:

```
/*proc "print" will print out to the output window*/
proc print data=SASINTRO.nations;
run;

/*proc "contents" will allows you to inspect the data in more details
(type, format, etc...)*/
proc contents data=SASINTRO.nations;
run;
```

To generate a table of basic descriptive statistics, such as the number of observations, mean, standard deviation, minimum, and maximum use the commands:

```
*all variables;
proc means data=SASINTRO.nations;
var ;
run;

*only two variables;
proc means data=SASINTRO.nations;
var corrup gdppc;
run;
```

SAS displays each variable name, along with its descriptive statistics in table form. To insert these results into another document, highlight the table, right-click, and select any of *copy* options. The results can now be pasted into another document.

Recode and Generate a new variable

Most of the times we are interested how the percentage change in one variable affected another variable. In those cases, we usually take the log of that variable. We will generate the log of GDP per capita *(gdppc)*. This can be done in Data Step.

```
data SASINTRO.nations; set SASINTRO.nations;
log_gdppc = log(gdppc);
run;
```

Making a frequency table of a continuous variable is not very useful. If we want a good frequency table, we should recode. Look at the description for the variable (*corrup*) you want to recode. This can be done using Freq procedure.

```
proc freq data=SASINTRO.nations;
tables corrup;
run;
```

We will recode this variable into three categories. Additionally, we will generate a new variable for the recoded data and provide labels for the new values in the Data Step.

```
data SASINTRO.nations; set SASINTRO.nations;
format corrup_categ;

if (corrup>=0) and (corrup<=3.3) then corrup_categ = 1;
else if (corrup>3.3) and (corrup<=6.6) then corrup_categ = 2;
else if (corrup>6.6) and (corrup<=10) then corrup_categ = 3;

label corrup_categ = 'Corruption Index in 3 categories';

proc format;
value corrup_categ
1 = 'High'
2 = 'Medium'
3 = 'Low'
;
run;

proc freq data=SASINTRO.nations;
tables corrup_categ;
run;
```

We can also assign a label to the variable itself to help us remember what it is. Format procedure is also helpful for formatting the values of the variables.

Finally, it is common to transform categorical variables into dummy variables in order to better capture the differences associated with being in one of those categories. To do this type:

```
DATA SASINTRO.nations; set SASINTRO.nations;

ARRAY dummys {*} 3.  corrup_categ1 - corrup_categ3;
DO i=1 TO 3;
dummys(i) = 0;
END;
if corrup_categ ne . then dummys(corrup_categ) = 1;

RUN;

PROC FREQ DATA=SASINTRO.nations;
TABLES corrup_categ*corrup_categ1*corrup_categ2*corrup_categ3/ list ;
RUN;
```

III. Creating Basic Graphs

In this section, we demonstrate how to generate a histogram and a scatter plot.

A histogram is used to show the frequency distribution of a variable.
Let's generate a histogram using the original variable and the log-transformation we just created (*gdppc*, *log_gdppc*)

```
proc univariate data=SASINTRO.nations;
var gdppc log_gdppc;
histogram;

Title 'GDP per Capita and Log-transformation';
run;
```

Suppose you are curious about the relationship between daily Log of GDP per capita and the Corruption(*corrupt*). Let's create a scatter plot to help visualize the relationship between these two variables. We will also include a trend line and title the graph to help us understand what we are seeing.

```
proc gplot data=SASINTRO.nations;

/* Define the axis characteristics */
axis1 label=("Corruption");
axis2 label=(angle=90 "Log of GDP per Capita") minor=(n=4);

/* Define the symbol characteristics for the scatter plot groups */
symbol1 interpol=none value=dot color=depk;
/* Define the symbol characteristics for the regression line */
symbol2 interpol=rl value=none color=black;

plot log_gdppc*corrup/haxis=axis1 vaxis=axis2;
plot2 log_gdppc*corrup/noaxis;

title1 'GDP per Capita and Corruption Relationship';
run;
quit;
```

As we can see, there is a positive relationship between these two variables.

IV. Bivariate Analysis

T-test

- A t-test is the most basic way to compare two groups together, based on their scores on a single variable. While we divided the **Corruption Index** into three groups, you can only compare two of those groups at a time using a t-test.
- Let's compare the countries with the most and least corrupted (group 1 and 3)

```
proc ttest data=SASINTRO.nations;
class corrup_categ;
var log_gdppc;
where corrup_categ~=2;
run;
```

- In the Output, look for the "t", which gives the value of the t-test, degrees of freedom, and the "p-value" or "Pr>|t|", which tells you the probability of getting that result. Remember that, in general, you are looking for a probability value less than .05.

Correlation

- The correlation coefficient is a measure of association between two variables. The sign indicates inverse versus directly proportional relationships. 0 would be a non-relationship, while a value of 1 would be a perfect positive relationship, and -1 would be a perfect inverse relationship.
- Let's look at the correlation between **Log of GDP per capita, Corruption Index**

```
proc corr data=SASINTRO.nations;
var log_gdppc corrup_categ;
run;
```

- Your output will be a 2x2 table, but only one cell is relevant. Use either cell that shows the relation between your two variables (there are two such identical cells).

V. Running Linear Regressions

In this section, we show you how to run a simple regression in order to understand the relationship between log of gdp per capita and corruption.

```
proc reg data=SASINTRO.nations;
model log_gdppc = corrup;
run;
```

SAS displays the regression results in table form. The important information provided is the Adj R-squared, coef. and t/P>|t|.

Let's run this regression again with robust standard errors by using procedure "robustreg"

```
proc robustreg data=SASINTRO.nations;
model log_gdppc = corrup;
run;
```